## REMARKS

Claims 1-41 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

**Section 102(b) Rejections:**

The Examiner rejected claims 1, 15-17, 27-29, 39 and 41 under 35 U.S.C. § 102(b) as being anticipated by Tran (U.S. Patent 5,864,689) (also referencing "Intel Architecture Software Developer's Manual (hereinafter "Manual"), and claim 40 as being anticipated by Carbine et al. (U.S. Patent 5,630,083) (hereinafter "Carbine"). Applicants respectfully traverse this rejection for at least the following reasons.

**Regarding claim 1, contrary to the Examiner's assertion, Tran fails to teach *a scheduler coupled to the dispatch unit and configured to schedule dispatched operations for execution, wherein in response to receiving a microcoded instruction, the dispatch unit is configured to dispatch to the scheduler a microcode subroutine call operation that includes a tag identifying a microcode subroutine associated with the microcoded instruction.*** The Examiner cites column 8, line 55 – column 9, line 4, and Tran's target address therein, as teaching that *in response to receiving a microcoded instruction, the dispatch unit is configured to dispatch to the scheduler a microcode subroutine call operation that includes a tag identifying a microcode subroutine associated with the microcoded instruction.* However, this citation does not describe Tran's microcode unit 45 and instruction decode unit 36 (which the Examiner equates with Applicants' dispatch unit) dispatching to a reservation station (which the Examiner equates with Applicants' scheduler) a microcode subroutine call operation including the limitations recited in claim 1. Instead, this citation describes, "when instruction decode unit 36 detects an instruction corresponding to a routine within microcode unit 45, an indication of the instruction is transferred upon instruction indication bus 62 to microcode unit 45" and also "the indication may comprise the target address of a subroutine call instruction." The Examiner appears to equate Tran's "target address" with the "tag" of

Applicants' claim 1. However, an indication of an instruction, even one that includes a target address of a subroutine call instruction (or tag), is not itself a microcode subroutine call operation, as recited in claim 1.

Furthermore, this citation describes that when instruction decode unit 36 detects a microcoded instruction, it sends an indication of the instruction to microcode unit 45. It does not describe dispatching a microcode subroutine call operation *to a scheduler*, as recited in claim 1. Note that according to Tran, the indication of the instruction is sent to microcode unit 45, not to Tran's reservation station (which the Examiner equates to the scheduler of claim 1). Tran's microcode unit 45 is not a scheduler. In fact, column 8, lines 61-64 goes on to describe that if instruction decode unit 36 detects an instruction to be performed by microcode unit 45, it stalls (i.e., stops dispatching instructions) until microcode unit 45 completes the corresponding routine.

In the Response to Arguments section of the Final Office Action, the Examiner submits:

> The execution of the x86 CALL instruction involves the storing of context information (such as the instruction pointer) in addition to the execution of the routine (microcode subroutine instructions in the context of Tran)... In order for context information to be stored, the x86 CALL instruction itself must be executed by one of the execution units of the processor. Execution by one of the execution units requires that the instruction be dispatched from decode unit (part of the dispatch unit) to the reservation station (i.e., the scheduler) associated with the desired execution unit.

Applicants note that the Examiner has cited no evidence that execution of the CALL instruction requires any of these specific actions by any of the specific components recited in Applicants' claims, as a CALL instruction may be implemented in a microprocessor using any of a wide variety of actions by various components of the microprocessor. In addition, the Examiner's remarks appear to **teach away from** the CALL instruction itself being a microcoded instruction for which the operations recited in claim 1 are performed. For example, the Examiner states that the "CALL instruction itself must be executed by one of the execution units," and that it is dispatched to the scheduler associated with the desired execution unit. In Tran, microcoded instructions

are not dispatched to one of the execution units 38 for execution, but are handled by microcode unit 45. This is clearly not the case for the execution of the CALL instruction itself, as noted in the Examiner's own remarks.

In addition, dispatching the CALL instruction itself to the scheduler of an execution unit 38, as the Examiner describes, is in direct contrast to the limitations of claim 1, in which *in response to receiving a microcoded instruction, the dispatch unit is configured to dispatch to the scheduler a microcode subroutine call operation*. The x86 CALL instruction, a source code instruction of the x86 instruction set architecture, is clearly not itself "a microcode subroutine call operation," but is a source-code level subroutine call.

Furthermore, contrary to the Examiner's assertions, the x86 CALL instruction is not an example of a microcoded instruction, as would be understood by anyone of ordinary skill in the art, and Tran does not teach that the CALL instruction itself is a microcoded instruction (i.e., one executed as in microcode). Instead, the CALL instruction is described as an example of a branch instruction, which also cannot be considered a microcoded instruction (see, e.g., claim 5, and column 4, lines 44 – 54, in which a standard subroutine call and return are described.) Tran teaches that if a branch type opcode (including a CALL opcode, for example) is detected and the target address of the branch is within a particular range of addresses, the target address may be routed to microcode unit 45. This is clearly not the same as a dispatch unit performing the functions recited in claim 1 (which are not taught by Tran) in response to receiving a microcoded instruction, as required by Applicants' claims. There is nothing in Tran that teaches a dispatch unit dispatching to the scheduler a microcode subroutine call operation in response to *receiving a microcoded instruction*.

Applicants remind the Examiner that anticipation requires the presence in a single prior art reference disclosure of each and every limitation of the claimed invention, arranged as in the claim. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The **identical** invention must

be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). As discussed above, Tran fails to disclose *in response to receiving a microcoded instruction, the dispatch unit is configured to dispatch to the scheduler a microcode subroutine call operation that includes a tag identifying a microcode subroutine associated with the microcoded instruction.* Therefore, Tran cannot be said to anticipate claim 1.

For at least the reasons above, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested.

Claim 17 includes limitations similar to claim 1, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 29, contrary to the Examiner's assertion, Tran fails to teach or suggest *detecting a microcoded instruction within the stream of instructions, wherein the microcoded instruction immediately precedes an other instruction in program order, in response to said detecting, dispatching a microcode subroutine call operation that identifies a microcode subroutine associated with the microcoded instruction.* The Examiner cites column 4, lines 36-41, as teaching detecting a microcoded instruction that immediately precedes another instruction in program order. However, this citation describes that if instruction decode unit 36 detects an instruction to be performed by microcode unit 45, it stalls (i.e., stops dispatching instructions) until microcode unit 45 indicates that the routine corresponding to that instruction has completed dispatch. This citation says nothing about the microcoded instruction immediately preceding another instruction in program order, as recited in Applicants' claim 29.

Further regarding claim 29, the Examiner cites column 8, line 55 – column 9, line 4, as teaching dispatching a microcode subroutine call operation in response to detecting the microcoded instruction. However, this citation does not describe dispatching a microcode subroutine call operation. In fact, as described above, Tran **teaches away from** the dispatch unit dispatching a microcode subroutine call operation in response to

detecting a microcoded instruction in the previous citation (instruction decode unit 36 stops dispatching instructions). Instead, this citation describes instruction decode unit 36 sending an indication of the instruction to microcode unit 45, which completes the routine. As discussed above regarding claim 1, sending an indication of an instruction is clearly not the same as dispatching a microcode subroutine call operation, as recited in Applicants' claim 29.

Finally regarding claim 29, the Examiner cites column 4, lines 42-54 as teaching that the microcode subroutine call operation pushes an address of the other instruction onto a stack, that the microcode subroutine includes a return operation, and that execution of the return operation pops the address from the stack (x86 Call instruction and x86 Return instruction). However, this citation has nothing to do with pushing or popping an instruction onto or off of a stack as part of a microcode subroutine call operation dispatched in response to detecting a microcoded instruction, as recited in claim 29. Instead it describes that source-code level subroutine call instructions (e.g., the CALL instruction of the x86 instruction set) having target addresses within a particular range of addresses are indicative of DSP functions. In this example, these DSP functions are executed by branching to a subroutine explicitly programmed using the CALL and RET instructions of the x86 instruction set. That is, the instruction detected by instruction decode unit 36 is itself a CALL instruction to a particular target address range, not a microcoded instruction for which a microcode subroutine call operation is dispatched in response to this detection. As discussed above, the x86 CALL instruction is not a microcoded instruction, but is an example of a branch-type instruction, which is detected in Tran.

In the Response to Arguments section of the Final Office Action, the Examiner argues that Tran has taught that a subroutine call instruction is dispatched in response to detecting a microcoded instruction. As discussed above, this is incorrect. The Examiner's citation in column 4, lines 47-52, describes a standard (source-code level) subroutine call and return, not a call and return from a microcode subroutine. This has nothing to do with whether or not a microcode subroutine performs push and pop

operations. In fact, in Tran, the dispatch unit stops fetching and dispatching operations while microcode unit 45 completes the microcode routine and then resumes fetching and dispatching operations in response to assertion of the signal complete line 64. Nothing in Tran discloses the microcode subroutine pushing and popping operations on a stack.

The Examiner further argues that, "the CALL instruction is a microcoded instruction as the operation to be executed in response to the instruction is, at least partially, contained within microcode... That is, the CALL instruction invokes microcode" and cites Tran, column 4, lines 36-54. First, the Examiner has provided no evidence for this definition of a microcoded instruction, nor is there any basis for this definition found in the cited art. Furthermore, the Examiner's characterization of these passages in Tran is incorrect. Column 4, lines 36 – 41 merely describes that instruction decode unit 36 stalls upon detection of an instruction to be performed by microcode unit 45, and column 4, lines 42 – 54 describe a standard (source-code level) subroutine call and return, not a microcoded instruction. Neither of these passages discloses that a CALL instruction can be considered a microcoded instruction as would be understood by one of ordinary skill in the art. As discussed above, a CALL instruction is an example of a branch-type instruction and is not a microcoded instruction, as required by Applicants' claim.

Finally, the Examiner submits that the x86 CALL instruction is both a microcoded instruction and a microcode subroutine call operation. This is clearly illogical and is not taught by Tran. In Tran, the CALL instruction is not a microcode subroutine call operation, nor is it dispatched to a scheduler in response to detection of a microcoded instruction. Instead, Tran teaches that a target address of a CALL instruction (which is also clearly not a microcode subroutine call operation) is sent to microcode unit 45 in response to detection of a branch-type instruction (the CALL instruction).

For at least the reasons above, the rejection of claim 29 is not supported by the cited art and removal thereof is respectfully requested.

Claim 41 includes limitations similar to claim 29, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 40, Carbine fails to teach or suggest *dispatching one or more operations included in a first microcode subroutine and one or more operations included in a second microcode subroutine, wherein said dispatching the one or more operations in the first microcode subroutine comprises performing register name replacements using replacement register names stored in a first alias table element and wherein said dispatching the one or more operations in the second microcode subroutine comprises performing register name replacements using replacement register names stored in a second alias table element.* The Examiner cites column 12, lines 35-56, as teaching generic microcode routines, and asserts that each of these generic microcode routines "has micro-alias registers" to replace register names within the routine. The Examiner seems to be implying that these generic routines, including what he interprets as separate micro-alias registers, teach dispatching operations from both a first and second microcode subroutine, and the use of both a first and second alias table element. However, there is nothing in this citation or elsewhere in Carbine that teaches multiple alias table elements or multiple micro-alias registers. Instead, Carbine describes a single micro-alias register (having multiple fields, but not multiple entries), which is loaded with different data dependent on which of four CUOPs is selected by multiplexer 560. There is also nothing in this citation that teaches dispatching multiple microcode subroutines, as recited in claim 40. Instead, Carbine describes microcode sequencing unit 534 issuing multiple CUOPs for one microcode flow (associated with a single entry point) at a time (see, e.g., column 11, lines 11-13). Since Carbine fails to teach or suggest dispatching operations from a first and second microcode subroutine and a first and second alias table element, Carbine cannot be said to anticipate claim 40.

In the Response to Arguments section of the Final Office Action, the Examiner submits that Carbine specifically refers to a plurality of micro-alias registers at column 12, lines 36-39. The Examiner also submits that Carbine states that registers are not hard-coded into any routine, "thereby indicating multiple generic microcode routines are

used. Therefore, Carbine has taught dispatching multiple generic microcode routines." Applicants assert, however, that even if multiple generic routines may exist in the system of Carbine, there is nothing that teaches dispatching two such routines. While the system of Carbine is directed toward parallel decoding of multiple instructions (see Abstract) there is nothing in Carbine that discloses dispatching operations from two different microcode subroutines, as required in Claim 40. Applicants assert that it is clearly not necessary that a system support dispatching operations from two different routines even if they may be decoded in parallel, and the Examiner has not cited anything in Carbine that discloses this limitation of Applicants' claim.

For at least the reasons above, the rejection of claim 40 is not supported by the cited art and removal thereof is respectfully requested.

Applicants also assert that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

**Section 103(a) Rejection:**

The Examiner rejected claims 2-6, 18-22 and 30-34 under 35 U.S.C. § 103(a) as being unpatentable over Tran in view of Carbine, claims 7, 8, 23, 24, 35 and 36 as being unpatentable over Tran and Carbine and further in view of Rotenberg, et al. ("Trace Cache: a Low Latency Approach to High Bandwidth Instruction Fetching") (hereinafter "Rotenberg"), claims 9, 10, 25 and 37 as being unpatentable over Tran in view of Kling (U.S. Publication 2004/0049657), and claims 11-14, 26 and 38 as being unpatentable over Tran and Kling and further in view of Harris (U.S. Patent 6,260,138). Applicants respectfully traverse these rejections. However, since the rejection of the independent claims has been shown to be unsupported, as discussed above, a further discussion of the dependent claims is not necessary at this time.

## CONCLUSION

Applicants submit the application is in condition for allowance, and prompt notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5500-81600/RCK.

Also enclosed herewith are the following items:

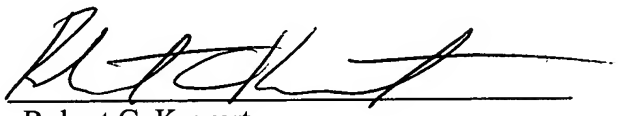
☒ Return Receipt Postcard

☐ Petition for Extension of Time

☐ Notice of Change of Address

☐ Other:

Respectfully submitted,



Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: September 14, 2006